TRACE-o1: RELIABLE SEARCH-AUGMENTED REASONING VIA STAGE-ALIGNED TRAJECTORIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Search-augmented reasoning enables LLMs to consult external evidence during inference, but search access alone does not guarantee reliable reasoning. Existing interleaved search–reasoning systems often make retrieval and reasoning decisions from local trajectory context, which can lead to fragmented evidence acquisition, accumulated intermediate errors, and unstable final predictions. We propose TRACE-o1, a stage-aligned framework that introduces reliability controls throughout the search-augmented reasoning process. Before reasoning, TRACE-o1 plans an evidence path to guide subsequent retrieval; during reasoning, it audits and repairs the evolving trajectory; and after reasoning, it stabilizes predictions across completed candidate trajectories through consistency-guided selection. Experiments across nine datasets show that TRACE-o1 improves overall answer accuracy and cross-trajectory stability relative to Search-o1-based baselines. Under a matched-$K{=}5$ comparison, TRACE-o1 improves macro accuracy from 60.5 to 63.6 over Search-o1 with self-consistency, with a 95% paired-bootstrap confidence interval of $[+1.4, +5.0]$ percentage points. These results support the benefit of combining evidence planning, trajectory repair, and consistency-guided selection within a unified framework.

1 INTRODUCTION

Large language models (LLMs) are capable general-purpose solvers, but their answers can remain unreliable on tasks requiring external, domain-specific, or recently updated knowledge. Retrieval-augmented generation (RAG) addresses this by grounding generation in retrieved evidence (Lewis et al., 2020; Mallen et al., 2023). Complex questions, however, often require multiple retrieval steps (Trivedi et al., 2023; Ammann et al., 2025). A reliable system may need to decompose the question, identify missing evidence, inspect retrieved information, and revise intermediate conclusions before producing a final answer (He et al., 2024).

Recent search-augmented reasoning systems address this need by interleaving generation with external search, letting models retrieve information during inference rather than only before generation (Yao et al., 2023; Press et al., 2023; Li et al., 2025). During generation, a model can issue a query, receive evidence, and continue reasoning with updated context, making evidence acquisition more adaptive than fixed pre-generation retrieval. Yet such loops often remain local and reactive: models search after detecting an information gap, but may not plan the evidence path, verify support for intermediate reasoning, or test final-answer stability across alternative completions.

This limitation highlights a central challenge in search-augmented reasoning: search access alone does not make reasoning reliable. A system may retrieve relevant evidence yet gather it piecemeal, misread retrieved information, accumulate local reasoning errors, or produce an answer that changes under a slightly different decoding path. These failures are especially visible in scientific, mathematical, and biomedical questions, where correctness often depends on effective evidence acquisition and disciplined reasoning control (Zhang et al., 2025; Han et al., 2024; Sohn et al., 2025).

We study how to improve the reliability of search-augmented reasoning without modifying the language model. Our key observation is that reliability failures occur at different points in a search–reasoning trajectory. Before reasoning, the system may lack a coherent evidence plan. During reasoning, it may make unsupported or inconsistent intermediate claims. Near the end of generation, it may arrive at several plausible but incompatible answers. A reliability-oriented framework should

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

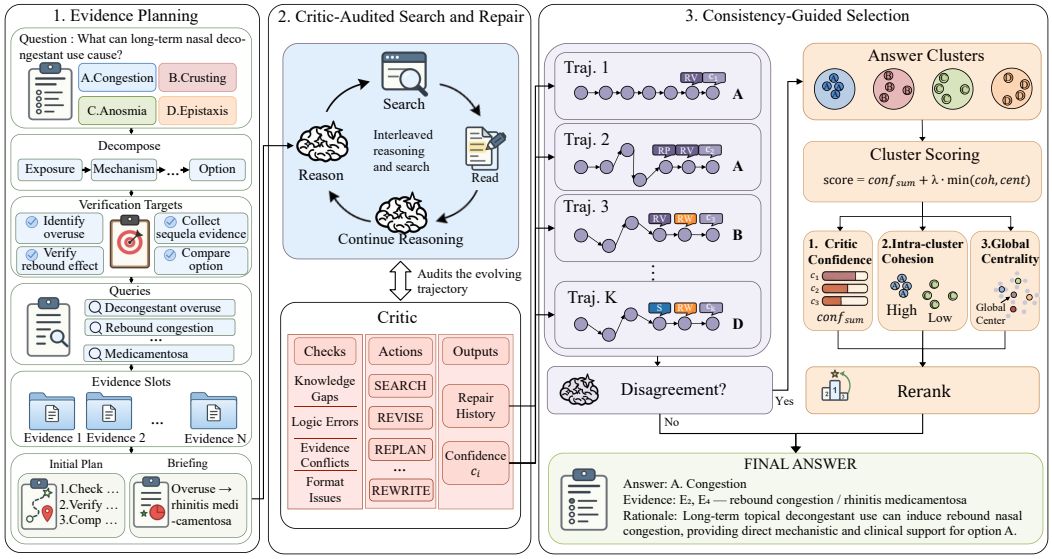

Figure 1: Overview of TRACE-o1. One plan and briefing initialize all K candidates; each then undergoes critic-guided search and repair before Consistency-Guided Selection aggregates them. S = Search, RV = Revise, RP = Replan.

therefore intervene where each failure arises, rather than expecting a single local search loop to address every source of error.

We propose TRACE-o1, a stage-aligned framework for improving the reliability of search-augmented reasoning trajectories. We use *reliability* to encompass more than final-answer accuracy: the system should acquire evidence deliberately, correct detectable process failures before they propagate, and remain stable under repeated candidate generation. Accordingly, TRACE-o1 places controls *before* reasoning (evidence planning), *during* reasoning (process-level critique and repair), and *after* completed trajectories (consistency-guided selection). The final stage first aggregates the completed candidates; it accepts the aggregate directly when no substantive disagreement is detected and invokes disagreement-aware reranking only for flagged cases. We further analyze cross-trajectory stability through candidate-answer disagreement, separating improvements in final accuracy from improvements in reasoning stability. Figure 1 illustrates the complete workflow.

Our contributions are as follows:

- We frame search-augmented reliability as a *stage-aligned* control problem and instantiate complementary interventions before, during, and after trajectory generation, rather than relying on a single reactive search loop. This formulation makes the location of each reliability failure explicit and links it to a corresponding intervention.

- We implement these controls as Evidence Planning, Critic-Audited Search and Repair, and Consistency-Guided Selection, targeting evidence acquisition, process-level error propagation, and cross-trajectory instability, respectively. The design therefore combines trajectory-local repair with cross-trajectory stabilization rather than treating the two as interchangeable controls.

- We evaluate TRACE-o1 on nine mathematical, general QA, and biomedical benchmarks and two open-weight backbones. Under the matched $K{=}5$ trajectory comparison, TRACE-o1 improves Search-o1+SC from 60.5 to 63.6 macro accuracy ($+3.1$ points), while reducing mean disagreement from 0.187 to 0.108; the full system reaches 63.6 macro and 45.1 micro accuracy. We additionally report paired-bootstrap confidence intervals for the matched trajectory and token-budget controls to quantify the robustness of these comparisons.

2 Related Work

Retrieval-Augmented Generation. Retrieval-augmented generation (RAG) augments parametric generation with external, non-parametric evidence (Lewis et al., 2020; Karpukhin et al., 2020; Islam et al., 2024), helping models handle long-tail or specialized knowledge that may be unreliable in model parameters alone (Mallen et al., 2023). Although surveys describe RAG as spanning naive, advanced, and modular pipelines (Gao et al., 2023), many systems still retrieve evidence only before generation or use a small fixed number of retrieval steps, despite recent progress on adaptive and domain-robust RAG (Jeong et al., 2024; Han et al., 2024; Sohn et al., 2025; Yoran et al., 2024). These pipelines can fall short on questions that require iterative evidence acquisition and reasoning repair. TRACE-o1 addresses this limitation by targeting unreliable search–reasoning trajectories where retrieval errors can persist into later reasoning, through stage-aligned planning, critique, and consistency controls, thereby treating reliability as a process-level property rather than a one-time retrieval outcome.

Search-Augmented and Tool-Augmented Reasoning. This limitation motivates methods that move beyond one-shot retrieval toward retrieval and tool use during generation. Prior work has explored active retrieval (Jiang et al., 2023), interleaved reasoning and actions (Yao et al., 2023), learned API use (Schick et al., 2023), question decomposition (Press et al., 2023), multi-hop evidence acquisition (Trivedi et al., 2023; Ammann et al., 2025), and agentic search workflows for long reasoning trajectories (Li et al., 2025). More recent agentic reasoning frameworks similarly emphasize tool use and structured control during reasoning (Ko et al., 2026). Rather than focusing only on when to search or how to inject retrieved evidence, TRACE-o1 focuses on improving the reliability of the entire search–reasoning trajectory as a unified control problem through stage-aligned evidence planning, process-level critique, and final-answer consistency, since errors may arise not only from missing evidence but also from poorly monitored intermediate reasoning.

Critique, Repair, and Consistency. A complementary problem is to critique, repair, and stabilize the reasoning trajectory after evidence acquisition. Self-RAG trains models to retrieve, generate, and critique with reflection tokens (Asai et al., 2024); Reflexion uses verbal feedback and episodic memory to improve decisions without weight updates (Shinn et al., 2023); and CRAG evaluates retrieval quality and applies corrections when evidence is unreliable (Yan et al., 2024). In TRACE-o1, the critic reviews the search-reasoning trajectory for evidence gaps, logical flaws, contradictions, and context-quality problems, making post-retrieval decisions more auditable, localizing faults before they propagate, and positioning critique as a trajectory-level control linked to work on verification, the limitations of self-critique, and iterative refinement (He et al., 2024; Stechly et al., 2025; Du et al., 2025). CoT methods elicit intermediate reasoning steps (Wei et al., 2022), recent work studies reasoning-trace verification and evaluation (Kim et al., 2026; Lee & Hockenmaier, 2025; Zhang et al., 2025), and self-consistency samples multiple reasoning paths to select the most frequent answer (Wang et al., 2023). TRACE-o1 applies this principle by generating multiple critic-audited search trajectories from the same plan and briefing, then aggregating normalized answers to reduce single-trajectory instability.

3 Method

Given question x, the system seeks final answer y with optional search-engine interaction. A search–reasoning trajectory is $\tau = (s_1, \ldots, s_T, y)$, where each step $s_t = (z_t, q_t, d_t)$ records reasoning state z_t, search query q_t, and retrieved evidence d_t when search is invoked. TRACE-o1 uses the standard interleaved search–reasoning interface and organizes reliability control into three stage-aligned modules in Figure 1: Evidence Planning, Critic-Audited Search and Repair, and Consistency-Guided Selection. Evidence Planning runs once; the audited search–repair stage then runs independently for each candidate before the third stage compares completed candidates for agreement and disagreement.

3.1 Evidence Planning

PLAN produces $P = (p, Q)$, where p concisely decomposes the question and $Q = (q_1, \ldots, q_m)$ is an ordered query sequence whose size is determined adaptively by the planner according to the question's information needs. The controller executes the resulting distinct queries and summarizes the retrieved

evidence into a compact background briefing B before trajectory generation. If initial retrieval is weak or empty, the implementation uses a retrieval judge to rewrite the query and retry retrieval; if no reliable evidence is found, the briefing remains empty rather than filled with unsupported snippets. Thus, planning supplies useful context without forcing unsupported evidence into the prompt. PLAN is invoked once per question, and the resulting (P, B) initializes all K candidate trajectories.

3.2 CRITIC-AUDITED SEARCH AND REPAIR

CRITIC acts as a process-level verifier over the question, current reasoning history, retrieved evidence, and provisional answer state. It checks for missing information, reasoning errors, truncation, and context-quality problems. When intervention is needed, it emits one of five repair actions—SEARCH, REVISE, REPLAN, RESTRUCTURE, or REWRITE—and otherwise returns NONE. Each action maps to a concrete handler on the active trajectory, so generation and repair can alternate inside the loop under critic supervision, instead of being driven only by the base model's local detection of evidence gaps. Detailed triggers and handlers are provided in Appendix C.

3.3 AUDITED CANDIDATE GENERATION

The audited search–repair pipeline is run independently K times from the same initial (P, B); candidates explore distinct search–reasoning paths; each candidate keeps a local copy (P_k, B_k) that can change only after REPLAN. For trajectory i, the final critic review yields an action a_i, feedback f_i, and a reliability score $c_i \in [0, 1]$, together with its repair history. Here, c_i is the critic's self-reported estimate of the reliability of the current trajectory and provisional answer given the available reasoning and evidence. It is not treated as a calibrated probability of correctness, but as an uncalibrated ranking signal for Consistency-Guided Selection. These critic-side signals are shown in Figure 1 and consumed by the final-selection stage. Algorithm 1 summarizes the end-to-end generation and selection procedure over K candidate trajectories.

Algorithm 1: TRACE-O1: Stage-Aligned Search and Selection

Require: Question x; candidates K
Ensure: Final answer $\hat{y}$
$P, B \leftarrow \text{PLAN}(x)$; $\mathcal{T} \leftarrow \emptyset$
for $k = 1$ to K **do**
  $P_k, B_k \leftarrow P, B$
  $\tau_k \leftarrow \emptyset$; $repaired \leftarrow$ **true**
  while $repaired$ and budget not exhausted **do**
    $\tau_k \leftarrow \text{SEARCHREASON}(x, P_k, B_k, \tau_k)$
    $(a, f, c) \leftarrow \text{CRITIC}(x, \tau_k)$
    $repaired \leftarrow (a \neq \text{NONE})$
    if $repaired$ **then**
      $(\tau_k, P_k, B_k) \leftarrow \text{REPAIR}(\tau_k, P_k, B_k, a, f)$
    end if
  end while
  $\mathcal{T} \leftarrow \mathcal{T} \cup \{(\tau_k, \text{NORMANSWER}(\tau_k), c)\}$
end for
$\mathcal{Y} \leftarrow \text{NORMANSWERS}(\mathcal{T})$
$(\tilde{y}, g, \rho) \leftarrow \text{LLMAGGREGATE}(x, \mathcal{T}, \mathcal{Y})$
if $g =$ **false then** ▷ no substantive disagreement
  return $\tilde{y}$
else
  return $\text{CLUSTERSELECT}(\mathcal{T}, \mathcal{Y})$
end if

3.4 CONSISTENCY-GUIDED SELECTION

The final stage treats the K completed trajectories as a gated consistency module and first normalizes their candidate answers $\mathcal{Y} = \{y_1, \ldots, y_K\}$. An LLM aggregator then reads the trajectories, normalized answers, and compact reasoning excerpts and returns an aggregated answer $\tilde{y}$, a review flag g (implemented as `need_posthoc`), and a candidate-consistency estimate ρ. If g is false, the module returns $\tilde{y}$ directly. If disagreement or uncertainty remains (g is true), it invokes disagreement-aware cluster scoring, combining answer-cluster support, critic-derived trajectory scores, and semantic consistency.

Each trajectory has a normalized answer y_i, critic-derived score c_i, and L2-normalized embedding e_i. For answer cluster $\mathcal{C}_a = \{i \mid y_i = a\}$, the final-stage score is

$$\text{score}(\mathcal{C}_a) = \text{conf_sum}(\mathcal{C}_a) + \lambda \min(\text{coh}(\mathcal{C}_a), \text{cen}(\mathcal{C}_a)) . \tag{1}$$

where $\text{conf_sum}(\mathcal{C}_a) = \sum_{i \in \mathcal{C}_a} c_i$. Given the normalized global mean $\bar{e} = \mu / \|\mu\|$ with $\mu = \frac{1}{K} \sum_{i=1}^{K} e_i$, the cohesion and centrality terms are

$$\begin{aligned} \text{coh}(\mathcal{C}_a) &= \tfrac{2}{|\mathcal{C}_a|(|\mathcal{C}_a|-1)} \sum_{\substack{i,j \in \mathcal{C}_a \\ i<j}} e_i^\top e_j, \\ \text{cen}(\mathcal{C}_a) &= \tfrac{1}{|\mathcal{C}_a|} \sum_{i \in \mathcal{C}_a} e_i^\top \bar{e}. \end{aligned} \tag{2}$$

We set $\mathrm{coh}(\mathcal{C}_a) = 0$ when $|\mathcal{C}_a| = 1$. Embeddings are computed after masking explicit final-answer expressions when applicable, reducing direct answer-string overlap and encouraging the semantic term to reflect reasoning similarity. The full decision rule is

$$\hat{y} = \begin{cases} \tilde{y}, & g = \textbf{false}, \\ \arg\max_a \mathrm{score}(\mathcal{C}_a), & g = \textbf{true}. \end{cases} \tag{3}$$

The aggregator consistency estimate ρ is retained as a diagnostic output and does not enter the cluster-score equation directly. The stage reuses the completed trajectories and introduces no additional retrieval or trajectory generation: every example incurs one aggregation call, while embedding-based cluster scoring is invoked only for examples flagged for post-hoc review.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate on nine benchmarks covering general QA, mathematical reasoning, and biomedical QA: Bamboogle (Press et al., 2023), GPQA (Rein et al., 2024), MuSiQue (Trivedi et al., 2022), AIME, AMC, Math500 (Hendrycks et al., 2021; Lightman et al., 2024), BioASQ (Krithara et al., 2023), PubMedQA (Jin et al., 2019), and MedQA (Jin et al., 2021).

Baselines. We compare with no-retrieval, multi-agent reasoning, search-interleaved, and retrieval-augmented methods. Baselines include Direct LLM, CoT (Wei et al., 2022), Vanilla MAD (Du et al., 2024), MAD-M^2 (Tian et al., 2026), MADC (Zhang et al., 2026), ReAct (Yao et al., 2023), Search-o1 (Li et al., 2025), Self-RAG (Asai et al., 2024), CRAG (Yan et al., 2024), Adaptive-RAG (Adapt-RAG) (Jeong et al., 2024), ReA-RAG (Lee et al., 2025), and MA-RAG (Wu et al., 2026).

Metrics and implementation. We report per-dataset accuracy in percentages, the unweighted macro-average across all nine benchmarks, and the dataset-size-weighted micro aggregate across all 5,407 examples. Macro and micro are computed from the unrounded per-dataset accuracies. The main comparison uses Qwen3-4B-Instruct-2507 (Yang et al., 2025; Qwen Team, 2025); a full cross-backbone evaluation with Llama-3.2-3B-Instruct (Meta, 2024; Grattafiori et al., 2024) is reported in Appendix A. All reported values use the same answer-normalization and evaluation pipeline. For both the matched-K=5 Search-o1+SC comparison and the stronger K=10 self-consistency control, we report 95% paired-bootstrap confidence intervals for the macro-accuracy difference. For each of 10,000 replicates, we resample paired questions with replacement within each dataset, compute the accuracy difference for each dataset, and then macro-average the nine differences with equal dataset weight. For the aligned Qwen3 multi-agent comparisons, we use five top-level agents/candidates and the same 32K runtime context envelope as TRACE-o1 while preserving each framework's native debate or retrieval-control logic. Ablation settings, prompts, and hyperparameters are provided in the appendix.

Contamination control. Because open-web search can surface benchmark mirrors, copied questions, or worked solutions, domain filtering alone cannot ensure that benchmark leakage is absent. As a first-line mitigation, the retrieval configuration blocks known benchmark- and dataset-hosting domains, including GitHub, Hugging Face, Kaggle, Papers with Code, Zenodo, and OpenReview. We do not treat this blacklist as sufficient evidence that no leakage occurred. Instead, we audit the *actual stored retrieval traces* for 1,273 questions spanning all nine benchmarks. Across 5,008 unique consumed records containing retrieval activity, manual review confirms exposure to question mirrors in eight audited questions, four of which also contain explicit solution cues. We therefore treat the audit as evidence of residual open-web contamination risk rather than proof that contamination is absent. Appendix E.3 provides the sampling protocol, retrieval-active coverage, matching rules, aggregate sensitivity analysis, and limitations.

4.2 Main Results

Table 1 reports the main Qwen3-4B comparison across three task families; full Llama-3.2-3B-Instruct results are deferred to Appendix A. TRACE-o1 raises the nine-dataset macro average from 59.1 to 63.6 and the dataset-size-weighted micro aggregate from 43.1 to 45.1 relative to single-trajectory Search-o1.

Table 1: Main Qwen3-4B-Instruct-2507 results across nine benchmarks. Best and second-best values are bolded and underlined, respectively.

Method	General QA			Math			Biomedical			Average	
	Bamb.	GPQA	MuSi.	AIME	AMC	M500	Bio.	PMQA	Med.	Macro	Micro
No Retrieval											
Direct LLM	29.6	57.1	5.3	23.3	75.0	67.8	34.3	75.4	68.7	48.5	37.3
CoT (Wei et al., 2022)	34.4	56.1	6.5	36.7	77.5	69.4	35.8	78.4	70.9	51.7	39.0
Multi-Agent Reasoning											
Vanilla MAD (Du et al., 2024)	33.6	61.6	5.7	63.3	90.0	71.6	38.0	73.2	75.6	57.0	40.1
MAD-M^2 (S) (Tian et al., 2026)	32.8	63.1	5.9	**70.0**	92.5	70.8	40.4	72.6	76.4	58.3	40.4
MAD-M^2 (O) (Tian et al., 2026)	35.2	63.1	6.7	**70.0**	95.0	72.2	39.2	75.6	75.8	59.2	41.1
MADC (Zhang et al., 2026)	35.2	63.6	6.1	63.3	95.0	74.8	37.0	73.0	76.2	58.3	40.7
Retrieval-Augmented Methods											
ReAct (Yao et al., 2023)	24.0	52.5	7.3	56.7	87.5	70.0	35.5	76.6	70.6	53.4	39.0
Self-RAG (Asai et al., 2024)	35.2	55.6	10.2	43.3	90.0	69.8	48.8	78.2	71.0	55.8	41.6
CRAG (Yan et al., 2024)	38.4	53.0	9.1	63.3	95.0	72.0	46.6	**81.4**	72.1	59.0	41.9
Adapt-RAG (Jeong et al., 2024)	30.4	53.5	2.9	43.3	82.5	71.2	41.4	76.4	70.7	52.5	37.6
ReA-RAG (Lee et al., 2025)	**40.8**	54.0	3.2	53.3	90.0	72.0	50.6	75.0	71.2	56.7	38.7
Search-o1 (Li et al., 2025)	34.4	58.1	9.4	53.3	95.0	79.0	50.6	77.4	74.4	59.1	43.1
MA-RAG (Wu et al., 2026)	33.6	61.6	5.8	**70.0**	95.0	**80.0**	37.3	74.4	**76.7**	59.4	41.3
TRACE-o1	**40.8**	**66.7**	**11.1**	66.7	**97.5**	**80.0**	**55.6**	79.0	75.0	**63.6**	**45.1**

More importantly, a matched-K comparison controls for the number of completed top-level trajectories and tests whether the gain can be explained solely by sampling more candidates: with $K{=}5$ for both systems, Search-o1+SC obtains 60.5 macro accuracy, while TRACE-o1 obtains 63.6.

Using the unrounded macro accuracies computed from the paired per-question predictions, the macro gain is $+3.1$ points, with a 95% paired-bootstrap confidence interval of $[+1.4, +5.0]$ over 5,407 paired examples. The methods can still differ in internal inference cost, which we report separately in Figure 2; Appendix B gives the bootstrap protocol and reproducibility details.

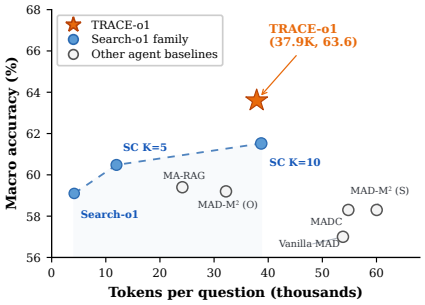

Figure 2: Macro accuracy–token trade-off.

Table 2: Matched-K trajectory comparison at $K = 5$.

Method	Macro	Mean $d\downarrow$	High-$d\downarrow$	Unanim. $\uparrow$
Search-o1+SC	60.5	0.187	27.73%	56.66%
TRACE-o1	**63.6**	**0.108**	**15.47%**	**74.56%**

This matched trajectory comparison also shows that the gain is not only a point-accuracy effect: mean disagreement decreases from 0.187 to 0.108, the high-disagreement rate falls from 27.73% to 15.47%, and the unanimous-answer rate rises from 56.66% to 74.56% (Table 2). These quantities characterize cross-trajectory stability under the common initial (P, B), rather than independence of candidate evidence acquisition. The aligned multi-agent baselines provide a complementary comparison: MA-RAG reaches 59.4 macro accuracy, MAD-M^2 (O) 59.2, MAD-M^2 (S) and MADC 58.3, and Vanilla MAD 57.0. On Llama-3.2-3B-Instruct, TRACE-o1 also achieves the highest macro accuracy among the evaluated methods (40.8), providing a cross-backbone check of the main result (Appendix A). Component-level effects are examined in the ablation study below; in particular, disagreement-aware reranking is treated as an internal substep of Consistency-Guided Selection rather than a separate module. CRAG and Self-RAG also perform strongly on individual benchmarks, while TRACE-o1 coordinates evidence preparation, process-level verification, and final stabilization throughout the search–reasoning trajectory. The two multi-hop open-form QA datasets (MuSiQue

and Bamboogle) remain harder for most systems and gain less from trajectory-level deliberation. This pattern may reflect limitations in partial-hop retrieval and the sensitivity of exact-match scoring.

Figure 2 compares nine-dataset macro accuracy with recorded token cost under the same Qwen3-4B setting. Token counts sum prompt and completion tokens across every logged LLM invocation, including planning and retrieval-related calls, trajectory generation, critique/repair, and final aggregation; external search requests themselves are reported separately rather than converted into tokens. Search-o1+SC at $K = 5$ spends nearly $3\times$ the tokens of single-trajectory Search-o1 for a modest gain, while several aligned multi-agent baselines consume more tokens than TRACE-o1 yet remain less accurate. TRACE-o1 reaches the highest macro accuracy in this comparison at 37.9K tokens per question. The reported TRACE-o1 token and call statistics include the final aggregation call, so Figure 2 and Table 3 use the same end-to-end accounting convention. Table 3 reports the corresponding calls, retrievals, and token statistics together with the token-matched Search-o1+SC control. Because these statistics aggregate all calls in the inference pipeline, they provide a common accounting view even when the internal call structure differs across methods.

4.3 Token-Budget-Matched Control

Matched candidate count does not by itself equalize inference cost. We therefore add a token-budget-matched Search-o1 control by scaling self-consistency to $K = 10$. This run records 38,708 tokens/Q, close to the 37,865 recorded tokens/Q of TRACE-o1, while preserving the Search-o1 reasoning/search policy and allocating the additional generation budget to independent trajectories. Because TRACE-o1 decomposes inference across planning, critique, repair, and aggregation stages, its call structure differs from Search-o1 and the model-call counts need not match exactly. We therefore do not interpret this experiment as a strict compute-matched comparison; rather, it tests whether simply allocating a comparable recorded end-to-end LLM-token budget to additional Search-o1 trajectories explains the improvement. This isolates extra candidate sampling while preserving the Search-o1 reasoning and retrieval policy. The control therefore changes the trajectory budget without introducing TRACE-o1's planning, critique, repair, or aggregation mechanisms, making the comparison more specific to the effect of additional sampling.

Table 3: Accuracy–cost comparison on Qwen3-4B-Instruct-2507. Costs are averaged per question.

Method	K	**Macro Acc.**	**Tokens/Q**	**Calls/Q**	**Tokens/Call**	**Searches/Q**
Search-o1	1	59.1	4,164	1.70	2,449	0.35
Search-o1+SC	5	60.5	11,971	11.96	1,001	1.75
Search-o1+SC	10	61.5	38,708	16.85	2,297	3.36
Vanilla MAD	5	57.0	53,807	10.34	5,202	0.00
MAD-M^2 (S)	5	58.3	60,037	15.34	3,914	0.00
MAD-M^2 (O)	5	59.2	32,200	10.34	3,115	0.00
MADC	5	58.3	54,799	10.00	5,480	0.00
MA-RAG	5	59.4	24,143	8.92	2,707	0.65
TRACE-o1	5	**63.6**	37,865	18.36	2,062	4.34

Interpretation. Scaling self-consistency to $K = 10$ increases Search-o1+SC to 61.5 macro accuracy while bringing its recorded token cost to the same range as TRACE-o1 (38,708 vs. 37,865 tokens/Q). We therefore treat this as a token-budget control rather than a strict compute-matched comparison. Search-o1+SC uses slightly fewer model calls (16.85 vs. 18.36 calls/Q) and fewer recorded search requests (3.36 vs. 4.34 searches/Q). Thus, this control is closely matched in recorded token budget but not in call or retrieval count, and we use it specifically to test whether additional Search-o1 sampling alone explains the accuracy difference. Under a similar recorded token budget, Search-o1+SC reaches 61.5 macro accuracy versus 63.6 for TRACE-o1, a $+2.0$ pp difference (95% paired-bootstrap CI: $[+0.5, +3.6]$ pp). Thus, even after increasing self-consistency to $K = 10$, the performance gap remains positive; simply sampling more Search-o1 trajectories does not account for the observed improvement. The different call and search profiles further show that similar recorded token budgets can correspond to different inference structures, so token matching should be interpreted as a budget control rather than as exact execution matching.

4.4 Ablation Study

Table 4 separates planning, critic-audited repair, and final consistency on GPQA, MedQA, and Math500. The full selection module combines cross-trajectory consistency with disagreement-aware reranking. In the table, "w/o" denotes the variant without reranking (RR), and bold/underlined values mark the best/second-best results.
The critic gives the largest single-component GPQA gain, while the consistency-only variant yields the strongest Math500 result among the three single-component settings. Planning and critique are complementary on GPQA. Final reranking raises GPQA from 65.7 to 66.7 and Math500 from 79.6 to 80.0, but lowers MedQA from 76.2 to 75.0. Representative planner behaviors appear in Appendix D.

Table 4: Ablation of stage-aligned controls.

Group	Crit.	Plan	Sel.	GPQA	MedQA	M500
Baseline	×	×	–	58.1	74.4	79.0
Single	✓	×	–	62.1	75.7	78.6
	×	✓	–	61.1	75.0	78.6
	×	×	Cons.	60.6	75.6	79.4
Pair	✓	✓	–	64.1	75.1	79.2
	✓	×	Cons.	63.6	<u>75.9</u>	79.4
	×	✓	Cons.	62.1	75.3	79.4
TRACE-o1 (w/o)	✓	✓	Cons.	<u>65.7</u>	**76.2**	<u>79.6</u>
TRACE-o1	✓	✓	**Cons.+RR**	**66.7**	75.0	**80.0**

4.5 Where Does TRACE-o1 Help?

To test whether TRACE-o1 provides larger gains on examples with unstable baseline predictions, we stratify GPQA using candidate-answer disagreement from five trajectories generated by the Baseline configuration in Table 4. For an input x, we define $d(x) = 1 - m(x)/5$, where $m(x)$ is the size of the modal normalized-answer cluster among the five candidates. Buckets are fixed before variant evaluation, with $d(x) \geq 0.4$ defining the high-disagreement subset (46 examples; 152 are low-disagreement).

Figure 3 confirms that the high-disagreement subset is harder: Baseline accuracy drops from 62.5% to 43.5%. Consistency-only and Critique-only are each -2.2 points relative to Baseline (one additional error among 46 examples), whereas Plan+Consistency and Critic+Consistency each gain $+6.5$ points and Plan+Critic gains $+13.0$.
The full diagnostic variant likewise gains $+13.0$ on the baseline-defined high-disagreement subset and $+7.2$ on the low-disagreement bucket; without reranking, the latter falls to $+5.9$. Appendix D reports exact bucket-wise accuracies; Figure 5 gives a complementary end-to-end AIME example.

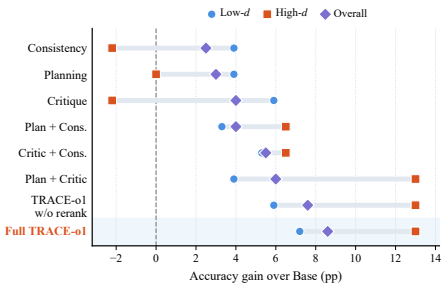

Figure 3: GPQA gains over Base (pp) by disagreement subset ($N = 152/46$).

4.6 Candidate-Count Diagnostics

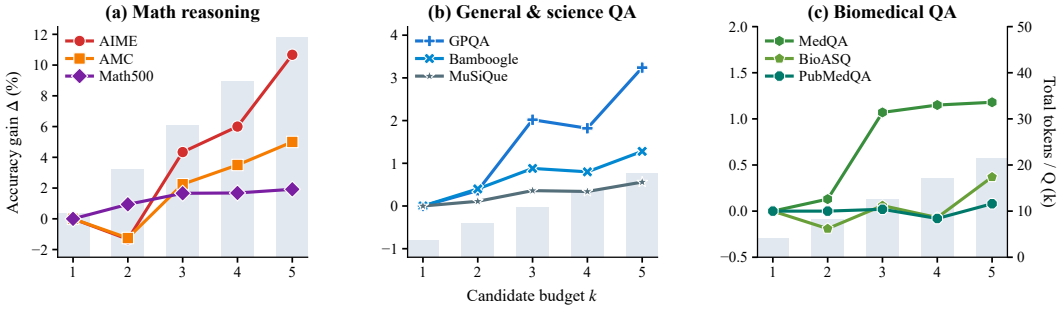

Figure 4: Per-dataset accuracy gain over k=1 as the candidate budget increases to k=5; gray bars show average token cost.

We vary k from 1 to 5 under simple majority voting to isolate the effect of adding completed trajectories; this is a fixed diagnostic baseline rather than the final TRACE-o1 selector. Figure 4 shows that the benefit is strongly domain-dependent. Math reasoning gains the most (AIME $+10.7$,

AMC +5.0, and Math500 +1.9 at k=5), while general/science QA shows a smaller but visible gain on GPQA (+3.2) and only minor movement on Bamboogle and MuSiQue. Biomedical QA shows limited additional benefit from increasing k, with changes of at most 1.2 points. The accompanying cost bars show the corresponding increase in total tokens as k grows, making the accuracy–cost trade-off explicit. We therefore use this result as a candidate-budget sensitivity diagnostic, rather than as evidence for the final selection mechanism itself.

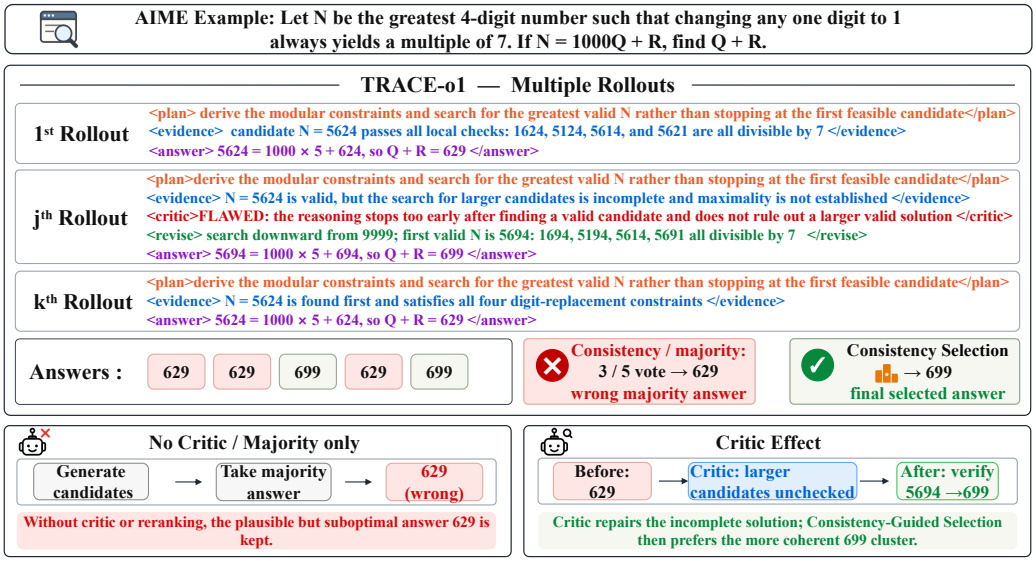

Figure 5: End-to-end AIME example: critic repair recovers 699 from a 629-majority candidate set.

Figure 5 makes the failure mode concrete. Several rollouts stop at $N = 5624$, which satisfies the digit-replacement checks but does not establish maximality; the raw $3/5$ majority therefore returns 629. In a competing rollout, the critic identifies this missing proof obligation, triggers revision, and continues the search to $N = 5694$, yielding 699. The final selection stage then selects this repaired minority candidate rather than the raw majority answer. The example illustrates the intended stage alignment: planning targets the greatest valid N, critique repairs premature stopping, and final selection retains the corrected solution.

5 Conclusion

We presented TRACE-o1, a stage-aligned framework for improving the reliability of search-augmented reasoning through evidence planning, process-level critique and repair, and consistency-guided final selection. Across nine benchmarks, TRACE-o1 improves the macro accuracy of the single-trajectory Search-o1 baseline from 59.1 to 63.6, while also improving cross-trajectory stability.

More importantly, our analyses suggest that the observed gains cannot be explained by candidate count or final aggregation alone. The three stages play complementary roles: planning structures evidence acquisition before trajectory generation, critique localizes and repairs failures during reasoning, and consistency-guided selection mitigates instability among completed trajectories. The disagreement-stratified and ablation results further indicate that the largest gains arise when these controls are combined, particularly on examples with high candidate disagreement under the baseline generation setting. Taken together, these findings motivate treating reliability as a process-level property of the entire search–reasoning trajectory rather than as a single retrieval or verification decision.

Several limitations remain. The additional reliability controls increase inference cost, and our evaluation is limited to two open-weight backbones and one retrieval backend. Moreover, final selection is fundamentally bounded by candidate quality: no selection rule can recover a correct answer that is absent from all generated trajectories. Improving candidate diversity, search efficiency, and adaptive allocation of reliability controls therefore provides a natural direction for future work.

Reproducibility Statement

We provide the implementation details required to reproduce TRACE-o1 in the appendix, including dataset splits, fixed run configurations, candidate and search budgets, decoding parameters, retrieval and query-handling rules, answer normalization, Consistency-Guided Selection, and the key prompt templates. Cross-backbone results and final-selection diagnostics are also reported with their corresponding settings. Appendix E collects the core reproducibility details used for the reported experiments.

AI Use Statement

Generative AI tools were used for language editing, organization, and general research and reference assistance. All scientific claims, experimental procedures, reported results, and citations were reviewed and verified by the authors, who remain responsible for the accuracy, originality, and academic integrity of the final paper.

References

Paul J. L. Ammann, Jonas Golde, and Alan Akbik. Question decomposition for retrieval-augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pp. 497–507, Vienna, Austria, 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.acl-srw.32/.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations*, 2024.

Chengyu Du, Jinyi Han, Yizhou Ying, Aili Chen, Qianyu He, Haokun Zhao, Haoran Guo, Sirui Xia, Jiaqing Liang, Zulong Chen, Liangyue Li, and Yanghua Xiao. Think thrice before you act: Progressive thought refinement in large language models. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/6882dbdc34bcd094e6f858c06ce30edb-Abstract-Conference.html.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 11733–11763. PMLR, 2024. URL https://proceedings.mlr.press/v235/du24e.html.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2023.

Aaron Grattafiori et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Rujun Han, Yuhao Zhang, Peng Qi, Yumo Xu, Jenyuan Wang, Lan Liu, William Yang Wang, Bonan Min, and Vittorio Castelli. RAG-QA arena: Evaluating domain robustness for long-form retrieval augmented question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 4354–4374, Miami, Florida, USA, 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.emnlp-main.249/.

Bolei He, Nuo Chen, Xinran He, Lingyong Yan, Zhenkai Wei, Jinchang Luo, and Zhen-Hua Ling. Retrieving, rethinking and revising: The chain-of-verification can improve retrieval augmented generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 10371–10393, Miami, Florida, USA, 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.findings-emnlp.607/.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *NeurIPS Datasets and Benchmarks*, 2021.

Shayekh Bin Islam, Md Asib Rahman, K S M Tozammel Hossain, Enamul Hoque, Shafiq Joty, and Md Rizwan Parvez. Open-RAG: Enhanced retrieval augmented reasoning with open-source large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 14231–14244, Miami, Florida, USA, 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.findings-emnlp.831/.

Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong Park. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7036–7050, Mexico City, Mexico, 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.naacl-long.389/.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, 2021. doi: 10.3390/app11146421.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W. Cohen, and Xinghua Lu. PubMedQA: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 2567–2577, 2019. doi: 10.18653/v1/D19-1259.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, Online, 2020. Association for Computational Linguistics. URL https://aclanthology.org/2020.emnlp-main.550/.

Minsu Kim, Jean-Pierre Falet, Oliver Richardson, Xiaoyin Chen, Moksh Jain, Sungjin Ahn, Sungsoo Ahn, and Yoshua Bengio. Latent veracity inference for identifying errors in stepwise reasoning. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/de89788c8105792bfb35a49d6dd4cae5-Abstract-Conference.html.

Dayoon Ko, Jihyuk Kim, Haeju Park, Sohyeon Kim, Dahyun Lee, Yongrae Jo, Gunhee Kim, Moontae Lee, and Kyungjae Lee. Hybrid deep searcher: Scalable parallel and sequential search reasoning. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=rXpTZyucal.

Anastasia Krithara, Anastasios Nentidis, Konstantinos Bougiatiotis, and Georgios Paliouras. BioASQ-QA: A manually curated corpus for biomedical question answering. *Scientific Data*, 10(1):170, 2023. doi: 10.1038/s41597-023-02068-4.

Jinu Lee and Julia Hockenmaier. Evaluating step-by-step reasoning traces: A survey. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 1789–1814, Suzhou, China, 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.findings-emnlp.94/.

Zhicheng Lee, Shulin Cao, Jinxin Liu, Jiajie Zhang, Weichuan Liu, Xiaoyin Che, Lei Hou, and Juanzi Li. Rearag: Knowledge-guided reasoning enhances factuality of large reasoning models with iterative retrieval augmented generation. *arXiv preprint arXiv:2503.21729*, 2025.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen-tau Yih, Tim Rocktaschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 2020.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. Search-o1: Agentic search-enhanced large reasoning models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 5420–5438. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.276.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/aca97732e30bcf1303bc22ac3924fd16-Abstract-Conference.html.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2023.

Meta. Llama 3.2 3B Instruct. https://huggingface.co/meta-llama/Llama-3.2-3B-Instruct, September 2024. Model card. Released September 25, 2024.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023.

Qwen Team. Qwen3-4B-Instruct-2507. https://huggingface.co/Qwen/Qwen3-4B-Instruct-2507, 2025. Model card.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, 2023.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023.

Jiwoong Sohn, Yein Park, Chanwoong Yoon, Sihyeon Park, Hyeon Hwang, Mujeen Sung, Hyunjae Kim, and Jaewoo Kang. Rationale-guided retrieval augmented generation for medical question answering. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 12739–12753, Albuquerque, New Mexico, 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.naacl-long.635/.

Kaya Stechly, Karthik Valmeekam, and Subbarao Kambhampati. On the self-verification limitations of large language models on reasoning and planning tasks. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/f3c5e56274140e0420baa3916c529210-Abstract-Conference.html.

Hongduan Tian, Xiao Feng, Ziyuan Zhao, Xiangyu Zhu, Rolan Yan, and Bo Han. Multi-agent debate with memory masking. In *International Conference on Learning Representations*, 2026.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022. doi: 10.1162/tacl_a_00475.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10014–10037, Toronto, Canada, 2023. Association for Computational Linguistics. URL https://aclanthology.org/2023.acl-long.557/.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022.

Wenhao Wu, Zhentao Tang, Yafu Li, Shixiong Kai, Mingxuan Yuan, Zhenhong Sun, Chunlin Chen, and Zhi Wang. From conflict to consensus: Boosting medical reasoning via multi-round agentic RAG. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=S7lpdz7NAW.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884*, 2024.

An Yang et al. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. Making retrieval-augmented language models robust to irrelevant context. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ZS4m74kZpH.

Qian Zhang, Jinyi Liu, Yan Zheng, Hebin Liang, and Lanjun Wang. Key decision-makers in multi-agent debates: Who holds the power? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 29883–29891, 2026. doi: 10.1609/aaai.v40i35.40235.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 10495–10516, Vienna, Austria, 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.findings-acl.547/.

A CROSS-BACKBONE RESULTS

Table 5: Full Llama-3.2-3B-Instruct cross-backbone results; notation and highlighting follow Table 1.

Method	General QA			Math			Biomedical			Average	
	Bamb.	GPQA	MuSi.	AIME	AMC	M500	Bio.	PMQA	Med.	Macro	Micro
No Retrieval											
Direct LLM	8.8	28.3	3.8	6.7	12.5	16.0	25.9	69.4	50.9	24.7	24.5
CoT (Wei et al., 2022)	28.8	26.3	4.8	13.3	22.5	15.2	28.1	76.4	54.4	30.0	27.0
Multi-Agent Reasoning											
Vanilla MAD (Du et al., 2024)	36.0	28.8	6.4	3.3	20.0	41.2	32.1	75.2	60.3	33.7	31.8
MAD-M^2 (S) (Tian et al., 2026)	36.0	23.7	6.4	3.3	25.0	44.6	30.9	73.0	60.2	33.7	31.7
MAD-M^2 (O) (Tian et al., 2026)	36.8	26.3	6.6	6.7	27.5	42.6	32.4	74.6	60.2	34.8	32.0
MADC (Zhang et al., 2026)	36.8	25.3	7.2	6.7	25.0	**44.8**	34.0	75.2	60.3	35.0	32.5
Retrieval-Augmented Methods											
ReAct (Yao et al., 2023)	24.0	23.7	7.1	6.7	17.5	32.0	42.3	73.6	58.2	31.7	30.8
Self-RAG (Asai et al., 2024)	24.8	28.3	8.3	13.3	**42.5**	39.2	44.4	77.2	56.2	37.1	32.4
CRAG (Yan et al., 2024)	24.8	29.3	9.7	**23.3**	35.0	38.8	45.7	77.6	55.7	37.8	33.0
Adapt-RAG (Jeong et al., 2024)	12.0	25.3	3.3	10.0	30.0	33.0	6.5	72.0	54.5	27.4	25.9
ReA-RAG (Lee et al., 2025)	29.6	28.3	9.6	13.3	35.0	33.2	**48.1**	74.6	54.3	36.2	32.0
Search-o1 (Li et al., 2025)	16.0	21.2	7.1	6.7	32.5	40.8	39.8	76.0	48.5	32.1	29.2
MA-RAG (Wu et al., 2026)	20.8	30.3	6.4	3.3	20.0	44.2	33.3	76.2	60.3	32.8	31.9
TRACE-o1	**37.6**	**32.3**	**9.9**	20.0	37.5	**44.8**	46.3	**78.2**	**60.4**	**40.8**	**35.2**

B PAIRED BOOTSTRAP UNCERTAINTY

We quantify uncertainty for both the primary matched-K comparison and the stronger self-consistency control using the exact per-question predictions underlying Search-o1+SC at $K = 5$ and $K = 10$, paired against the same TRACE-o1 predictions. Questions are paired by normalized question text within each dataset. In each of 10,000 bootstrap replicates, we resample paired questions with replacement within each dataset, compute the accuracy difference per dataset, and then take the equal-weight macro average across the nine datasets. This preserves the paper's dataset-level macro definition rather than letting larger datasets dominate.

For the matched-$K = 5$ comparison ($N = 5{,}407$), TRACE-o1 improves macro accuracy over Search-o1+SC by $+3.08$ pp, with a 95% CI of $[+1.40, +4.96]$ pp. For the stronger Search-o1+SC $K = 10$ control, the improvement remains $+2.04$ pp, with a 95% CI of $[+0.49, +3.62]$ pp. The main text rounds these values to one decimal.

C CRITIC REPAIR ACTIONS

Table 6: Repair actions emitted by CRITIC during audited search.

Action	Diagnostic Trigger	Trajectory Repair
NONE	No issue is detected.	Continue the current trajectory.
SEARCH	Evidence is missing or insufficient.	Issue an additional retrieval step.
REVISE	A reasoning step is unsupported or invalid.	Inject targeted feedback into the reasoning chain.
REPLAN	The evidence plan is inadequate.	Regenerate the evidence plan.
RESTRUCTURE	Context is noisy or poorly organized.	Reorganize reasoning history or retrieved evidence.
REWRITE	The prompt or provisional answer is malformed.	Rewrite the prompt or provisional answer state.

D PLANNING ANALYSIS

D.1 DISAGREEMENT-STRATIFIED GPQA ANALYSIS

Table 7: GPQA accuracy by disagreement subset ($N = 152/46$); Δ is relative to Base.

Condition	Low Acc.	Low Δ	High Acc.	High Δ
Base	62.5	–	43.5	–
Consistency	66.4	+3.9	41.3	-2.2
Planning	66.4	+3.9	43.5	+0.0
Critique	68.4	+5.9	41.3	-2.2
Plan+Con.	65.8	+3.3	50.0	+6.5
Critic+Con.	67.8	+5.3	50.0	+6.5
Plan+Critic	66.4	+3.9	56.5	+13.0
TRACE-o1 w/o rerank	68.4	+5.9	56.5	+13.0
Full (diagnostic)	69.7	+7.2	56.5	+13.0

Table 7 provides the exact bucket-wise values behind the main-text disagreement analysis in Figure 3.

D.2 QUALITATIVE TRAJECTORY ANALYSIS

Table 8 illustrates when critic repair is effective. Cases 60 and 119 expose local chemistry and algebra errors, while Case 69 requires structural re-derivation. In each successful case, the critic points to a checkable constraint and the revision changes only the affected reasoning step. Case 125 is deliberately retained as a failure mode: the critic detects that the predicted product is incompatible with the options, but repeated critic–replan loops cannot resolve the apparent mismatch between the predicted product and the available options. Together, these cases support using critique as a localized repair mechanism rather than as a replacement solver; failures that cannot be repaired locally require candidate diversity or stronger external validation.

Table 8: Representative critic-repair cases and outcomes.

Case	**Error type**	**Wrong reasoning**	**Critic feedback**	**Revised reasoning / outcome**
60	Regiochemistry error	Predicts para substitution: "4-bromo."	Points to the meta position: "3-bromo."	Corrects the product to 3-bromonitrobenzene.
119	Algebraic derivation error	Uses $(1/\mathrm{plx})^2(1/\mathrm{plx}^2) = 1/\mathrm{plx}^3$.	Notes that the product is $1/\mathrm{plx}^4$.	Corrects the exponent and derives $dN \propto -\mathrm{plx}^{-4}\, d(\mathrm{plx})$.
125	Option-set inconsistency	Predicts enantiomers plus a meso form.	Argues that the reaction should predominantly yield meso-2,4-pentanediol.	Remains unresolved: the plausible product is absent from the option set, causing critic–replan loops.
69	Structural symmetry error	Assigns $D_{\infty h}$ symmetry to E by analogy with C.	Requires determining E's symmetry from its molecular structure.	Recomputes E directly and selects C_{2v}.

D.3 REPRESENTATIVE PLANNING CASES

Table 9 merges the planner analysis with the representative case set. Across the four examples, the planner instantiates task-specific evidence structures rather than merely rewriting the question: it converts numerical questions into formula-driven workflows (Case C1), enumerates ORF and frameshift checks for sequence verification (Case C2), organizes mechanistic evidence around the key reaction step (Case C3), and reframes diagnostic questions as audits of hidden workflow failures

(Case C4). This is most useful when the required evidence path is hidden, but the planner is not a verifier; incorrect intermediate inferences still require critic feedback.

Table 9: Representative planner outputs across four task types.

Case	Question	Plan Content	Search Queries	Category / Function
C1	Two quantum states have lifetimes of 10^{-9} s and 10^{-8} s; what energy difference is sufficient to resolve them?	(1) Establish that level resolution requires the energy gap to exceed the lifetime-induced linewidth. (2) Apply the energy–time uncertainty relation $\Delta E \approx \hbar/\Delta t$. (3) Compute the linewidth for each state. (4) Compare the larger linewidth with the answer options.	energy–time uncertainty principle and energy linewidth calculation; uncertainty principle for quantum energy levels and lifetime relation; resolving two energy levels in quantum systems based on lifetime; minimum energy difference for resolving two quantum states with specified lifetimes	Numerical reasoning Formula selection → parameter substitution → option comparison
C2	Select the plasmid that can correctly express full-length human P53.	(1) Compare each plasmid DNA sequence with the target P53 coding region. (2) Verify the ATG start codon and the complete ORF. (3) Detect premature stop codons, frameshifts, and incorrect start sites. (4) Select the plasmid with an uninterrupted coding sequence.	human P53 protein amino acid sequence to DNA sequence conversion with the ATG start codon; comparison of plasmid DNA sequences with the human P53 open reading frame; identification of the correct start codon and uninterrupted ORF in P53 expression plasmids; common errors in expression plasmids, including frameshifts and premature stops	Sequence / ORF verification Target definition → candidate comparison → anomaly check
C3	Determine the products of a pinacol–pinacolone rearrangement problem.	(1) Identify the governing reaction mechanism. (2) Analyze carbocation formation after protonation and water loss. (3) Determine the migrating group in the 1,2-shift. (4) Compare the predicted products with the answer options. (5) Verify migratory aptitude, especially aryl versus alkyl migration.	pinacol rearrangement mechanism with vicinal diols and carbocation migration; migratory aptitude in pinacol rearrangement for aryl versus alkyl groups; pinacol rearrangement of substituted vicinal diols	Mechanism-driven reasoning Mechanism identification → key-step analysis → product verification
C4	Identify the issues that most commonly cause subtle but severe errors in genomics data analysis.	(1) Reformulate the task as a diagnosis of hidden failure sources in genomics workflows. (2) Evaluate each candidate issue according to prevalence and impact. (3) Cross-check the candidates against reported bioinformatics pitfalls and best practices. (4) Select the most plausible hidden error source.	common error sources in genomics data analysis caused by reference genome mismatch; chr versus no-chr confusion in genomic coordinate systems; impact of incompatible data formats on genomics pipelines and error detection	Diagnostic reasoning Checklist construction → error-source audit → failure localization

E REPRODUCIBILITY DETAILS AND PROMPTS

This appendix records the fixed experimental recipe used for the reported TRACE-o1 runs. Modules are enabled or disabled by the experimental variant, and the full setting enables planning, critic-audited repair, and Consistency-Guided Selection. The third stage always adds one LLM aggregation call; disagreement-aware cluster scoring over the completed trajectories is performed only when the aggregator emits `need_posthoc=true`. The stage does not issue new searches, repairs, or trajectory generations.

E.1 FIXED RUN CONFIGURATION

Table 10 summarizes the fixed implementation settings used for the reported TRACE-o1 runs and final-selection diagnostics.

Table 10: Fixed implementation settings for the reported TRACE-o1 runs.

Component	**Setting used for reproduction**
Backbones	Qwen3-4B-Instruct-2507 for the main block and Llama-3.2-3B-Instruct for the cross-backbone block, served through an OpenAI-compatible vLLM endpoint.
Data splits	GPQA uses the `diamond` split; all other datasets use `test`.
Candidate budget	`num_candidates`=5 completed trajectories per question. The candidate-count diagnostic evaluates prefixes $k = 1, \ldots, 5$.
Search budget	`max_search_limit`=10, `max_turn`=5, retriever `top_k`=10, and `max_doc_len`=3000 characters per retrieved document context.
Retrieval backend	Tavily-compatible web search with `engine_type` =`tavily`, `scraper_type=auto`; cached runs use the same normalized result schema of URL, title, snippet, and optional context.
Contamination-control filter	Blacklist known benchmark- and dataset-hosting domains (GitHub, Hugging Face, Kaggle, Papers with Code, Zenodo, and OpenReview); the retrospective audit below checks residual ordinary-site mirrors and filter escapes.
Retrieval leakage audit	Audit 1,273 questions across all nine benchmarks using stored retrieval traces and benchmark-aware mirror, answer-key, and worked-solution rules; details are in Appendix E.3.
Initial planner retrieval	The planner runs once per question and adaptively determines a task-dependent set of distinct search queries; per query, an LLM selects at most one URL from up to five candidates; the resulting plan and briefing are reused by all candidates.
Generation decoding	Main reasoning: temperature 0.7, top-p 0.8, top-k sampling 20, and `max_tokens`=32768. Webpage-to-reasoning analysis uses temperature 0.7 and top-p 0.8.
Planner decoding	Planning, URL selection, extraction, synthesis, and replan temperatures are 0.7. Their top-p values are 0.9, 1.0, 1.0, 0.9, and 0.9, respectively.
Critic and repair decoding	Critic temperature 0.7 and critic top-p 0.5.
Consistency-selection embeddings	BAAI/bge-m3 embeddings through the SiliconFlow OpenAI-compatible embeddings endpoint, with L2-normalized vectors and a SHA1-keyed embedding cache.
Consistency-selection score	The final-selection coefficient is selected on the independent 9-task non-test calibration in Appendix F.1 and then fixed to $\lambda = 8$ for all reported test evaluations: $\mathrm{ConfSum} + \lambda \cdot \min(\mathrm{coh}, \mathrm{cen})$. `ConfSumOnly` denotes the critic-score-sum-only ablation.

E.2 QUERY HANDLING RULES

Planner-generated queries are parsed from `search_queries` in planner order. The planner determines the number of queries adaptively according to the information needs of each question, and the controller executes the resulting distinct queries in planner order. Search queries emitted by the main reasoner must appear between `<|begin_search_query|>` and `<|end_search_query|>`; the implementation strips surrounding whitespace, ignores empty queries, and does not execute an exact repeated query within the same trajectory. A critic `SEARCH` action may add one additional query through the same execution path. A critic `REPLAN` action reruns the planner only for that trajectory and updates its local plan/briefing.

E.3 Open-Web Retrieval Contamination Audit

Open-web retrieval creates a validity threat because benchmark questions and solutions may appear on ordinary educational websites even after known dataset-hosting domains are blocked. We therefore conduct a question-level audit of the *actual stored retrieval traces*. The audit covers 1,273 distinct questions across all nine benchmarks, using complete benchmark sets where feasible and fixed-seed subsets otherwise.

The primary audit unit is a retrieval-bearing record consumed by the reasoning pipeline, including selected search results and fetched evidence. We exclude the later `webpage_analysis` summaries from mirror detection because those summaries are model-generated from the retrieved evidence and can restate the user question. Table 11 therefore also reports the number of *retrieval-active* questions—questions with at least one primary consumed record—so that the search coverage of each historical trace is explicit.

We distinguish three leakage signals. *Question mirror* requires a long normalized overlap beginning near the benchmark question; *answer-key leakage* requires the gold answer in an answer/solution context; and *worked-solution leakage* requires a question mirror together with explicit solution cues. For multiple-choice benchmarks we match option text rather than option letters, while isolated answer mentions in open-form QA are treated as ordinary retrieved evidence unless mirror and solution cues co-occur. All automatic flags are manually reviewed.

Table 11: Aggregate open-web contamination audit.

Metric	Count
Audited questions	1,273
Retrieval-active questions	494
Unique consumed records	5,008
Question mirrors	8
Answer-key-only leakage	0
Worked-solution leakage	4

Table 11 summarizes the aggregate audit across all nine benchmarks, separating audit coverage (questions and consumed records) from confirmed leakage incidence.

Manual review identifies 8 confirmed question-mirror cases, of which 4 also contain worked-solution leakage; no answer-key-only case is confirmed. We additionally evaluate accuracy sensitivity on the same stored $K{=}5$ audit bundle. Excluding flagged questions, or conservatively scoring all flagged questions as incorrect, does not materially change the qualitative accuracy pattern. These are retrospective sensitivity diagnostics on the stored audit bundle, not filtered-retrieval reruns or replacements for the main-table scores.

We also retain two secondary diagnostics. The historical traces contain five primary records from blacklist-domain escapes; two overlap confirmed mirrors, while three do not contain the target question or answer and are therefore not counted as contamination. Among 3,209 stored pre-selection candidate snippets, 20 snippets across 11 questions trigger the mirror/solution heuristics. These snippets are visible to URL selection but are not themselves primary reasoning evidence, and snippet storage is incomplete for some historical runs.

The audit remains conservative in scope: deterministic matching can miss heavily paraphrased copies, and search was not activated for every audited question, as reflected by the retrieval-active total. Conversely, source-title or topical overlap alone is not treated as benchmark leakage. We therefore view the audit as a measured sensitivity analysis and evidence of residual open-web contamination risk, not proof that contamination is absent.

E.4 Answer Normalization and Gating

After generation, NormAnswer is deterministic. It extracts the final answer from the last `\boxed{...}` span, supporting nested braces. For choice and QA modes, an inner `\text{...}` wrapper and surrounding parentheses are removed. Generated-answer tasks are lowercased and whitespace-normalized. QA tasks additionally remove punctuation and the articles “a”, “an”, and “the”. Choice tasks map option letters, option text, and common aliases to a canonical answer key. PubMedQA maps yes/no/maybe directly, maps true/false to yes/no, and maps uncertainty phrases such as “insufficient” or “cannot determine” to maybe. Other unmapped outputs are marked invalid; no such cases occur in the reported PubMedQA candidate bundle.

After per-trajectory normalization, TRACE-o1 first applies the LLM aggregation prompt in Figure 8. If the aggregator sets `need_posthoc=false`, its aggregated answer is returned directly. If it sets `need_posthoc=true`, TRACE-o1 applies the consistency-guided cluster score described in Appendix F.2. This gated selector uses the same completed candidate trajectories and adds no retrieval, repair, or new trajectory generation. Majority voting is retained only as a simple diagnostic baseline for analyzing final-selection behavior.

E.5 Consistency-Guided Cluster Scoring

Consistency-Guided Selection groups completed trajectories by exact normalized answer; empty or missing answers become `UNK`. For answer cluster $\mathcal{C}_a$, the critic-score term sums the final critic trajectory-reliability scores c_i of its member trajectories; these are used as uncalibrated ranking signals rather than probabilities of answer correctness. For semantic scoring, candidate text removes explicit final-answer strings such as `\boxed...`, `final answer`, and option echoes, lowercases text, collapses whitespace, and keeps the final 1200 characters. BGE-M3 embedding vectors are L2-normalized.

For each cluster, coh is the mean pairwise dot product among vectors inside the cluster; singleton clusters use 0.0. cen is the mean dot product between cluster vectors and the global candidate center. The primary selector scores

$$\text{score}(\mathcal{C}_a) = \sum_{i \in \mathcal{C}_a} c_i + \lambda \cdot \min(\text{coh}(\mathcal{C}_a), \text{cen}(\mathcal{C}_a)).$$

Ties are resolved by cluster size, then center score, then mean candidate score, and finally the normalized answer string.

F Final-Selection Sensitivity and Validity Diagnostics

F.1 Independent Non-Test Calibration of the Final-Selection Coefficient

To select the final-selection coefficient without tuning on reported test examples, we construct a frozen task-aligned non-test calibration set with 50 examples per task (450 questions total). Native training splits are used after removing overlap with the reported test set; where no native training split is available, we use non-overlapping task-aligned pools (pre-test-year AIME/AMC, MATH training problems for Math500, and HotpotQA training questions as a multi-hop proxy for Bamboogle).

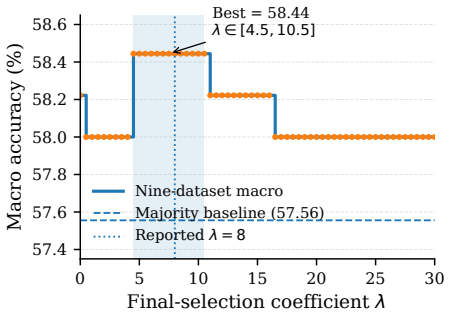

Figure 6: Independent non-test calibration of λ on 450 task-aligned questions. The sweep exhibits a broad maximum-accuracy plateau, and we select $\lambda = 8$ as a representative interior value.

TRACE-o1 generates the $K = 5$ trajectories once, and the sweep changes only λ in $\text{ConfSum} + \lambda \min(\text{coh}, \text{cen})$ using the answer-masked BGE-M3 representation. Holding the completed trajectories fixed isolates the effect of the post-hoc selection coefficient from retrieval, critique, repair, and candidate generation. Of the 450 questions, 124 exhibit candidate disagreement and can therefore be affected by re-selection. Figure 6 shows a broad performance plateau rather than a sharp optimum, indicating that the selector is not highly sensitive to small changes in λ around the best region. We therefore select $\lambda = 8$ as a representative interior value of this plateau and fix it for all reported test evaluations.

F.2 Final-Selection Variants

Table 12 summarizes the final-selection rules used in the diagnostic comparison. In TRACE-o1, the LLM aggregator is always applied first. Its answer is accepted directly when `need_posthoc=false`; disagreement-aware cluster scoring is entered only when `need_posthoc=true`. The variants below

compare deployable filtering or re-ranking rules for this post-hoc branch, using the same completed trajectories. We evaluate all available completed trajectories for GPQA, Math500, and MedQA. Table 13 reports the macro-average lift of each selector over the majority baseline on these three evaluation sets. Variants use the same completed candidate trajectories and add no retrieval or new trajectory generation, so accuracy gaps are attributable to the selection rule rather than generation or search budget.

Table 12: Final-selection variants used in the diagnostic comparison.

Selector	**Simple description**
Majority (baseline)	Votes over normalized candidate answers; ties broken by CONFSUM.
HeuristicConfSumMin	Critic-score sum plus semantic-consistency reward.
EmbeddingCohesionAll	Chooses the most internally coherent embedding cluster.
EmbeddingCriticCohesionAll	Mixes critic weighting with embedding cohesion.
SafeCurrent	Keeps the current answer unless its critic score is low.
ConsensusThenWeighted	Uses consensus first, then critic-weighted voting.
CriticWeighted	Weights votes by critic-derived scores.
FilteredWeighted	Filters weak candidates before weighted voting.
EmbeddingCenterAll	Chooses the answer closest to the embedding center.
BestCritic	Selects the highest-scoring critic candidate.
ConfidenceOnly	Historical selector name; selects the highest critic-score candidate.

Table 13: Macro-average selector lift over majority on GPQA, Math500, and MedQA (pp).

Selector	**Δ vs. majority (pp)**
HeuristicConfSumMin	+0.42
EmbeddingCohesionAll	+0.11
EmbeddingCriticCohesionAll	+0.08
ConsensusThenWeighted	+0.00
SafeCurrent	+0.00
CriticWeighted	-0.03
FilteredWeighted	-0.08
EmbeddingCenterAll	-0.89
ConfidenceOnly	-1.43
BestCritic	-1.43

Three patterns are visible in Table 13. First, selectors that combine multiple weakly correlated signals can yield modest gains over majority voting, with HeuristicConfSumMin giving the largest improvement. Second, every selector that commits to a single trajectory based on one feature alone—ConfidenceOnly, BestCritic, or the embedding-centroid variant—underperforms the majority baseline in this diagnostic. Third, the spread between the best and worst variants is roughly 1.85pp within the same fixed-trajectory setting; we therefore treat the TRACE-o1 selector as a final-stage stabilization rule rather than a primary driver of accuracy.

F.3 RETROSPECTIVE GPQA ROBUSTNESS CHECK

As an additional diagnostic, we vary only the semantic-reward coefficient in the final selector on a fixed GPQA $K = 5$ candidate set, while keeping the same questions, completed trajectories, normalized answers, critic-derived scores, and BGE-M3 semantic features fixed. Thus the analysis requires no new retrieval, repair, or trajectory generation. Accuracy changes only modestly across the examined range and exhibits a broad plateau rather than an isolated optimum. Because GPQA is itself a reported evaluation set, this analysis is retained strictly as a retrospective robustness check and is not used for coefficient selection. The coefficient used in the reported evaluations is fixed by the independent non-test calibration in Appendix F.1.

G KEY PROMPT TEMPLATES

Placeholders such as {QUESTION}, {HISTORY}, and {CONTENT} are filled at runtime.

Prompt stack for TRACE-o1 (PLAN)

```
PLAN

**Task:**
You are an expert research planner. Given the user's question, you need to:
1. Analyze the question to understand the core information needs.
2. Create a step-by-step plan to answer the question.
3. Generate a small set of specific, effective search queries sufficient to gather the
     necessary information. Determine the number of queries adaptively based on the question's
     information needs, and avoid redundant queries.

**User Question:**
{QUESTION}

**Output Format (JSON only):**
```json
{
  "plan": "Brief step-by-step plan...",
  "search_queries": [
    "query 1",
    "query 2"
  ]
}
```
```

Prompt stack for TRACE-o1 (CRITIC)

```
CRITIC

You are a strict academic reviewer and editor.

User Question: {QUESTION}
Current History: {HISTORY}
Finish Reason: {FINISH_REASON}

Checklist:
1. **Truncation**: Was the output truncated? (finish_reason == 'length')
2. **Knowledge Gap**: Is there missing information? If yes, suggest a specific search query.
3. **Logic**: Is the reasoning sound?
4. **Context Quality**: Is the context repetitive, contradictory, or poorly structured?

Output Format (JSON only):
```json
{
  "status": "OK or FLAWED",
  "action": "NONE or SEARCH or REVISE or REPLAN or RESTRUCTURE or REWRITE",
  "query": "If action is SEARCH, provide query; else empty",
  "feedback": "Concise critique",
  "restructured_context": "If action is RESTRUCTURE, provide reorganized context; else empty",
  "rewritten_prompt": "If action is REWRITE, provide new prompt; else empty",
  "confidence": 0.0
}
```
The confidence field estimates the reliability of the current trajectory and
provisional answer given the available reasoning and evidence: 0 means highly
unreliable and 1 means highly reliable. It is a self-report used for ranking,
not a calibrated probability of correctness and not confidence in the critic's
diagnosis itself.

Actions:
- NONE: No issues found
- SEARCH: Need to search for missing information
- REVISE: Minor correction needed, continue with feedback
- REPLAN: Need to redo background research
- RESTRUCTURE: Reorganize context structure (remove duplicates, fix contradictions)
- REWRITE: Rewrite the entire prompt with a summary (for truncation or severe issues)
```

Figure 7: System prompts for planning and critique; the critic targets substantive search, reasoning, and context-quality issues.

Prompt stack for TRACE-o1 (CONSISTENCY AGGREGATOR / REVIEW FLAG)

```
AGGREGATOR

You are aggregating five independent candidate trajectories for the same question.
Decide the most reliable final answer and whether another post-hoc selector should review it.
Return ONLY valid JSON. Do not wrap it in markdown fences.

JSON schema:
```json
{
  "answer": "YOUR_FINAL_ANSWER",
  "need_posthoc": true,
  "consistency_score": 0.0,
  "reason": "short reason"
}
```

Rules:
- consistency_score must be a number from 0 to 1.
- Higher consistency_score means the candidates agree and your aggregation is reliable.
- Set need_posthoc=true when answers conflict, reasoning quality is mixed, or you are uncertain.

- Set need_posthoc=false only when the candidate answers and reasoning are consistent enough.
- {answer_rule}

Question:
{question}

Candidate trajectories:
Candidate {candidate_index}:
Extracted answer: {extracted_answer}
Reasoning excerpt:
{reason_excerpt}

...

Return the JSON now:
```

Figure 8: LLM aggregation prompt. The need_posthoc flag gates disagreement-aware cluster scoring.